(4)

Technical Report 1284
April 1989

Cost Metric Algorithms For Internetwork Applications

D. Olsen
R. Dillard

Approved for public release; distribution is unlimited.

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152–5000

E. G. SCHWEIZER, CAPT, USN
Commander

R. M. HILLYER
Technical Director

ADMINISTRATIVE INFORMATION

This project was performed for the Independent Exploratory Development Office of the Naval Ocean Systems Center. The work was carried out by System Design and Architecture Branch (Code 854) of the Naval Ocean Systems Center, San Diego, CA 92152-5000.

Released by
M. B. Vineberg, Head
System Design and
Architechture Branch

Under authority of
E. R. Jahn, Head
Battle Force and Theater
Communications Division

BP

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NOSC TR 1284	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Ocean Systems Center 6b. OFFICE SYMBOL (if applicable) Code 854	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) San Diego, CA 92152-5000	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of Chief of Naval Research 8b. OFFICE SYMBOL (if applicable) OCNR-20T	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code) Arlington, VA 20217	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62936N PROJECT NO. ZE68 TASK NO. RV31612 AGENCY ACCESSION NO. DN308 065

11. TITLE (include Security Classification)

COST METRIC ALGORITHMS FOR INTERNETWORK APPLICATIONS

12. PERSONAL AUTHOR(S)

D. Olsen and R. Dillard

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM TO	April 1989	49

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	multinetwork controller Unified Networking Technology System (UNT) advanced technology demonstration "C" Language Production System (CLIPS)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The particular function of the Multi-Network Controller (MC) to be implemented for the 1990 Unified Networking Technology Advanced Technology Demonstration addressed in this Independent Exploratory Development (IED) project is the final selection of a subnet in response to a Transmit Service Request. The architectural description of the MC defines databases and performance measures used by this IED effort. In FY88, initial conclusions about the applicability of neural networks, fuzzy set methods, cost/value functions, and expert systems were investigated and documented. A Real Time Expert System (RTES), using subroutines that implement the decision techniques described, was selected as the best method for experimentation. A transportable, embeddable RTES shell (CLIPS) was chosen, and implementation of subnet selection algorithms began.

A follow-on IED effort for FY89 has been funded. An independent research effort for FY89 is also a spin-off of the FY88 IED effort, and is described in this report.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE PERSON D. Olsen	22b. TELEPHONE (include Area Code) (619) 553-4970 22c. OFFICE SYMBOL Code 854

DD FORM 1473, 84 JAN

83 APR EDITION MAY BE USED UNTIL EXHAUSTED
ALL OTHER EDITIONS ARE OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

SUMMARY

OBJECTIVE

The particular function of the Multinetwork Controller (MC) to be implemented for the 1990 Unified Networking Technology (UNT) Advanced Technology Demonstration (ATD) addressed in this Independent Exploratory Development (IED) project is the final selection of a subnet in response to a Transmit Service Request. The architectural description of the MC defines databases and performance measures used in this IED effort.

RESULTS

In fiscal year 1988, initial conclusions about the applicability of neural networks, fuzzy set methods, cost/value functions and expert systems were investigated and documented. A Real Time Expert System (RTES), using subroutines that implement the decision techniques described, was selected as the best method for experimentations. A transportable, embeddable RTES shell – C Language Production System (CLIPS) – was chosen, and implementation of subnet selection algorithms began.

RECOMMENDATIONS

A follow-on IED effort for fiscal year 1989 has been funded. An independent research effort for fiscal year 1989 is also a spin-off of the fiscal year 1988 IED project and is described in section 7 of this report.

CONTENTS

FIGURES

TABLES

1. INTRODUCTION

Plans for future Naval communications place a heavy emphasis on the interconnection (internetworking) of communications resources. Internetworking enhances the robustness and expediency of communications systems. Multiple problems rise from this goal. One such problem is the development of algorithms enabling an internetwork gateway to select subnets for the routing of various data types and priorities. The purpose of this Independent Exploratory Development (IED) project is to propose and analyze algorithms addressing this subnet selection problem. (The larger problem of developing internetwork routing algorithms is addressed in a follow-on IR project, see section 7.2). These algorithms were to be developed for implementation in the Unified Networking Technology (UNT) 1990 Advanced Technology Demonstration (ATD), while remaining applicable to a general internetwork gateway.

Several theoretical solution approaches were investigated: neural networks, fuzzy decision methods, cost/value functions, and expert systems. The results of these investigations are presented below, prefaced by overviews of the UNT Project, the Multinetwork Controller (MC) (the internetwork gateway developed for UNT), and the Final Network Selection algorithm, where the solution algorithms will be put to the test. Several of the sections reference data structures and algorithms are used in the MC Architecture document (D. Olsen, 1988).

2. BACKGROUND

2.1 OVERVIEW OF THE UNT PROJECT

The following sections are taken from the UNT Concept section of the *Advanced Technology Demonstration Unified Networking Technology Phase 1 Program Plan* (Code 854, 1987).

The overall concept of UNT is to develop techniques which will allow communication systems to function as a resource that can be dynamically allocated, rather than the present fixed dedicated links. Furthermore, the dynamic allocation should be transparent to the users requiring communication services. UNT, in general, covers all aspects of Naval Communications as summarized in figure 1. The diverse group of users shown in figure 1 is typical of existing Navy systems and will probably remain so even with the potential introduction of Local Area Network (LAN) technology. UNT must develop interface standards which will allow this large group of diverse users a common access to UNT resources, without forcing major changes in the USER systems, both present and in the future.

The three major functions to be evaluated are the Multinetwork Controller (MC), Network Administrator (NA) and Link Controller (LNC). The term functions is used to indicate that UNT is basically the development, test and evaluation of algorithms that can be used or distributed in a variety of processors. UNT is not the development of new hardware.

The MC function concentrates on the interface control to the various users, the selection of the best available data link, and the flow control of traffic between the two.

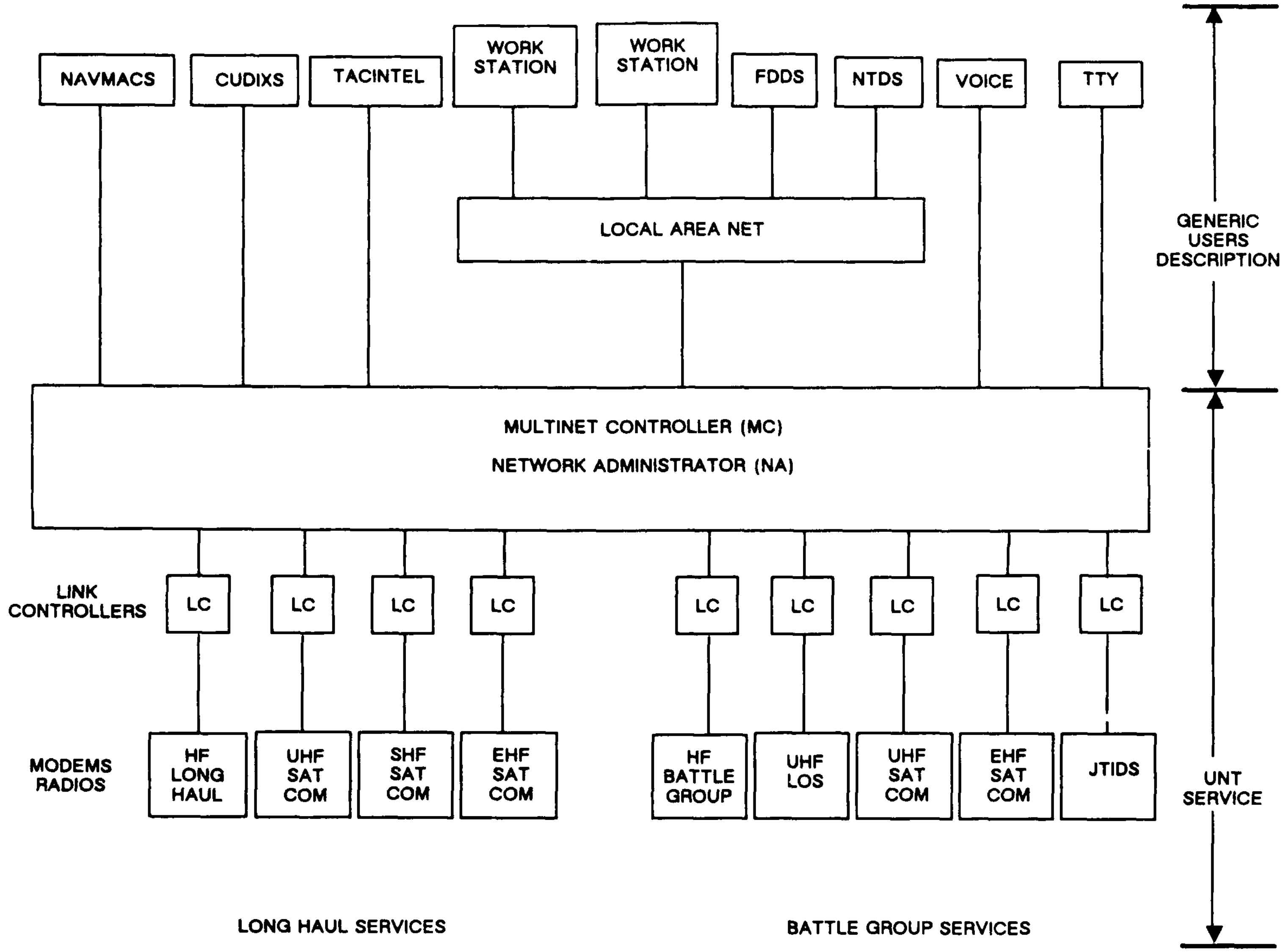

Figure 1. UNT system architecture.

The UNT concept is targeted for an advanced technology demonstration in 1990. Seven UNT nodes will be fielded: five land sites, one surface platform, and one air platform. Each node will host a UNT equipment suite consisting of several data sources, an HF subnet equipment suite, a UHF subnet equipment suite, an MC, a LAN, and interface units connecting the components to the LAN.

2.2 OVERVIEW OF THE MULTINETWORK CONTROLLER

The MC performs an enhanced gateway function. It potentially interconnects multiple radio frequency (RF) subnets (although only two RF subnets will be available for the UNT ATD). The MC monitors and collects performance statistics on the RF subnets. The MC controls access to the RF resources, granting or denying requests from various data sources to utilize the RF resources. The decision whether to grant permission for a data source to utilize an RF resource (or resources), and which RF resource(s), is the major function of the MC. This decision is carried out by the Network Selection Algorithm (NSA), a four step algorithm. The purpose of each step is highlighted below. The fourth step, final network selection, is the focus of this IED project.

For the purposes of this investigation, it was assumed that a number of specified RF subnet performance parameters were available to the MC.

2.3 OVERVIEW OF THE NETWORK SELECTION ALGORITHM

The following sections are taken from the MC Operational Overview section of the Multinetwork Controller Architecture Specification (D. Olsen 1988).

2.3.1 Introduction

Choosing the best network(s) to fill a given transmit service request is a balance between physical, practical, and heuristic parameters that are not easily compared to each other. For example, a satellite network may currently be supplying fast, reliable service to a message's intended destination. Obviously, it is the best network to carry the message, if the evaluation criteria is performance. However, the message may be very low priority--for example, it may be a request for four tons of chocolate ice cream. It may not be preferable to send such a message over the satellite network; perhaps the communications officer wants to keep the network clear for high priority messages. In this example, the performance capability of the network is called a heuristic parameter, and the network preference is called a practical parameter.

Some criteria take on a mixture of physical, practical, and heuristic parameter characteristics. This adds another dimension of complexity when attempting to select a network or networks; from what viewpoint is the criteria evaluated? A very important example is connectivity. From one viewpoint, the question Is the source node and destination node connected by the network? imposes an absolute criteria on the network selection process. If they aren't connected, the network is not a feasible choice. At the same time, the connectivity question is a heuristic problem; the source and destination may be connected with some probability (relays may be fading in and out of connection range, the subnet may be experiencing intermittent partitioning due to jamming, etc.).

Since the network selection problem does not lend itself to a numerical evaluation or comparison of the various criteria involved, a structured, multiphase approach to the problem is suggested. The approach described here uses physical criteria to weed out unfeasible networks, dynamic criteria to identify available networks, assigns a preference class to each subnet to introduce communications policy into the algorithm, computes heuristic parameters (measures of subnet performance), and applies a rule based evaluation of different metrics to make a final decision.

Four basic steps must be addressed by any algorithm attempting to choose the best network(s) over which to transmit a message regardless of the special characteristics and needs of the message or the complexity of the algorithm:

(1) Identify feasible, available networks.

(2) Consider policy defined network preference criteria.

(3) Make an estimate of expected network performance.

(4) Make the final network selection.

Each of these steps is discussed in detail.

2.3.2. Step 1a. Identify Feasible Networks

A feasible network is one which is physically capable of transmitting a message. Examples of feasibility criteria are Does the network supply the necessary bandwidth? Do the destination site(s) have the necessary equipment to be connected to the network? Does the network support the transmission mode needed by this message? These kind of criteria are called absolute because a network failing to deliver the minimum required level of service is absolutely incapable of transmitting the message and must be deleted from consideration. The output from Step 1a is a candidate set of networks for a specific transmission.

2.3.3. Step 1b. Delete Unavailable Networks

A network may be unavailable for several reasons: the LNC serving the network is down, current EMCON conditions specify the RF frequencies used by the network are banned, current network use, and To be Determined (TBD). The difference between feasibility and availability is that feasibility criteria are constant, while availability criteria are dynamically changing with current conditions and use of communication media. For example, an HF network currently dedicated to a voice circuit is unavailable as a candidate for record message or tactical data transmission (until the voice circuit is closed). Similarly, EMCON conditions could specify no transmissions below Super High Frequency (SHF), causing all HF, UHF (and other) networks to be unavailable. The output from Step 1b is a set of currently available networks (a subset of the Step 1a candidate networks).

2.3.4. Step 2. Determine Network Preference

Any feasible, available network is physically capable of transmitting the message, however, the current Communications Plan (COMMPLAN) must be considered when making a network selection. The COMMPLAN is a communications policy developed by Battle Force command, specifying primary, secondary, tertiary, etc., preferences for different kinds of traffic of given priority to be delivered on particular networks. The COMMPLAN may reserve or limit access to certain networks, or designate certain networks as preferable for certain types of message traffic. This step may be viewed as a multilayered filtering of the feasible networks. The result is zero or more feasible, available networks, with an attribute assigned to each network indicating its preference. Assigning a value to this attribute is equivalent grouping the networks into sets. These sets will be designated Preference Classes. The networks within each Preference Class are equal in their (COMMPLAN determined) desirability as transmission candidates.

If a network is not assigned a Preference Class for a specific data type, it is eliminated by this step. For example, the MILSTAR network might be designated as a primary network for command level voice traffic, and as a secondary network for priority 7 and priority 8 NAVMACS messages. If a request was made for transmission of an NTDS message, MILSTAR would be eliminated from consideration. Alternatively, the COMMPLAN might dictate that MILSTAR be the last alternative for record messages, with high preference going to command level voice traffic. In this case, MILSTAR would not be totally filtered out; it would be classed into the lowest Preference Class.

2.3.5. Step 3. Apply Network Behavior Prediction Algorithms

Observe that the first two steps do not address two very important issues connectivity and timeliness. Connectivity is the issue of whether a connected route actually exists on the network from the source to the destination. Timeliness is the issue of how long will it take the network to deliver the message (correctly) to the destination(s). Both of these issues are affected by many criteria: congestion on the network, number of relays that must be made to deliver a message on the network, overall network throughput, electromagnetic interference (adverse atmospheric disturbances, hostile jamming), etc.

These are not static criteria that may be entered into a table, or set by policy; they are dynamic, changing conditions within a network. Not even the cleverest algorithm can correctly model these elements all the time. However, they are of extreme importance in selecting a network for transmission of a message. Traffic load balancing, timely transmission of priority messages, and internetwork routing are several of the factors affected by the timeliness and connectivity models.

Step 3 implements probabilistic models of increasing complexity to estimate expected network performance. The performance parameters estimated are dependent on the Build. Performance estimates are passed to step 4 for use in making the final network selection.

2.3.6. Step 4. Final Network Selection

Steps 1 to 3 reduce the network selection problem to a manageable size by deleting unfeasible and unavailable candidate networks, then assigning Preference Classes and estimated performance attributes to the remaining candidates. The problem of the final selection still remains. At what point is it decided that a preferred network's estimated performance is insufficient, and less preferable networks be considered? There are two basic alternatives:

(1) Present the steps 1-to-3 results to a system operator, who makes the decision.

(2) Implement some criteria to make an automatic, intelligent decision.

The major drawback to automation is the complexity to implement a really intelligent algorithm. The algorithm can be as simple as an alternating selection between networks of the highest preference (as in Build 0) or as complex as an artificial intelligence solution.

This IED project's focus is the automation of Step 4. Given a set of specified decision criteria for each available, feasible subnet, how can one or more be selected that will supply the best possible service for the specified data under the current operational environment? This is a substantial problem; there is no one best solution. Several theoretical approaches are investigated; an integrated solution is chosen for the 1990 ATD that utilizes a real time expert system and value functions.

3. APPLICABILITY OF NEURAL NETWORK METHODS

NOSC Code 421 (C. Priebe, D. Marchette) was tasked to investigate the applicability of neural network techniques to the network selection problem. Their recommendation was that neural networks could be used to resolve portions of the problem (but only portions that are easily solved by conventional means); the general problem lies outside the strengths of neural network techniques.

One of the biggest strengths of neural networks is their ability to learn to supply reasonable answers to situations not previously encountered. To do so, most neural networks must be presented with a training set of real data coupled with the desired response. Confronted with a data set never previously encountered, the neural network presents an answer that is close in some metric to the patterns learned from training data.

This is potentially disastrous in our application. Data sets representing different Transmit Service Requests (TSRs) might be virtually identical, yet require widely different responses. A good answer from the neural network might not be a valid answer in terms of the communications environment. Further, a neural net would not be capable of isolating those parameters which, if relaxed slightly, would enable a TSR to be granted.

Despite these problems, the neural network approach will be kept in the background with an eye towards further development. When decision criteria and metrics are completely defined, it may be that neural networks can be applied to this problem in such a way that validity of solutions is ensured.

4. APPLICABILITY OF FUZZY DECISION METHODS

Network selection decisions should be based on a number of different criteria, and the precise criteria can be defined in many different ways for each selection problem. Suppose, for example, that we specify three of our criteria to be the following: (1) probability of delivery, (2) time to delivery, and (3) balance of network loading. These criteria are by no means mutually independent. Dempster-Shafer decision methods require independent evidence (or criteria, in this case), and Bayesian methods become unmanageable with this degree of mutual dependence. Fuzzy decision methods, however, do not require independence and are appropriate for this kind of problem.

Some decision methods involve assigning importance weights to the criteria. The methods of assigning weights originate primarily from the psychology community. The use of the weights is generally (if not always) ad hoc. Logarithmic opinion pooling (Genest and Zidek 1986) is a quasi-Bayesian way of using weights. A number of different ways of employing weights in fuzzy decision algorithms have been proposed in the literature over the past dozen years or so, and two examples of these will be described in the next section. Several recent methods of using weights involve ordered weighted averaging, and are beyond the scope of our discussions. The fuzzy decision methods described next are known as Max-Min methods and are among the simplest.

4.1. MAX-MIN FUZZY DECISION METHODS

The notation we will use for describing Max-Min fuzzy decision methods is as follows.

Given: A set of n alternatives, A = (a1,...,ai,...,an),

a set of m criteria, and

sets C1,...,Cj,...,Cm, where Cj = (cj1,...,cji,...,cjn).

Cj is a fuzzy subset of A, where the measure cji indicates how well alternative ai satisfies the jth criterion. The notation "^" indicates the intersection (the logical AND) of two fuzzy sets.

Figure 2 illustrates the simplest Max-Min method, proposed by Bellman and Zadeh (1970). Figure 3 shows the Max-Min method extended to use weights that relate to the importance of the criteria (Saaty 1977, Yager 1977).

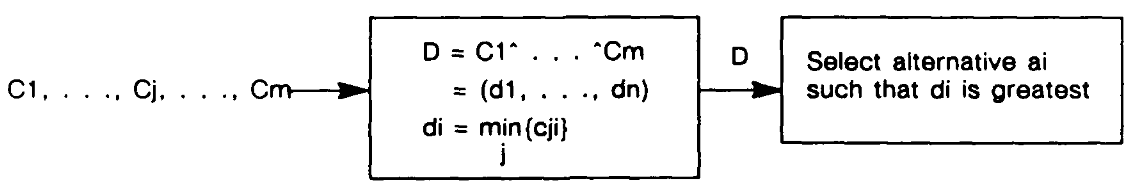

Figure 2. Bellman/Zadeh (1970) Max-Min decision method.

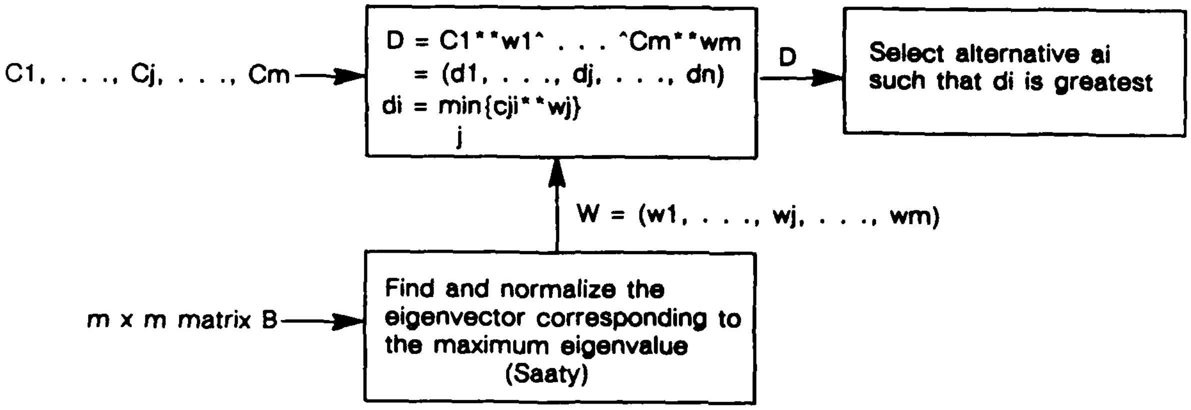

Figure 3. Saaty(1977)/Yager(1977) version of the Max-Min method.

The elements of Saaty's scaling matrix B shown in figure 3 are found by asking the expert to judge the relative importance of each criteria over the others. For example, b31=3 means that criteria 3 is judged to be three times more important than criteria 1. The highest comparison values is 9, which is interpreted to be absolute importance, one over the other. The comparison value 1, of course, is equal importance. The matrix B has the following form:

$$
\begin{array}{c}
\text{crit.} \\ 1 \\ 2 \\ \cdot \\ \cdot \\ k \\ \cdot \\ \cdot \\ m
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} 1 & 2 & \dots & j & \dots & m \end{array} \\
\begin{bmatrix}
1 & 1/b21 & \dots & 1/bk1 & \dots & 1/bm1 \\
b21 & 1 & \dots & & & \\
\cdot & & & & & \\
\cdot & & & & & \\
bk1 & & \dots & & & \\
\cdot & & & & & \\
\cdot & & & & & \\
bm1 & & \dots & & & 1
\end{bmatrix}
\end{array}
$$

bkj = integer between 1 and 9
bjk = 1/bkj.

To find the weights, one solves the eigenvalue problem:

$$BY = (\text{max eigenvalue})Y.$$

One then obtains the eigenvector (y1,...,ym) corresponding to (max eigenvalue), and lets wj = m*yj / sum yj. Normalization is optional.

Figure 4 illustrates a variation of a Max-Min method proposed by Yager (1981). A measure S is used both for scaling the importance of the criteria and for measuring the grade of membership of alternative ai in Cj.

$$
\begin{aligned}
S &= \text{(none, very low, low, medium, high, very high, perfect)} \\
&= (s0, s1, s2, s3, s4, s5, s6) = \{sk\} \\
&= (0, 1, 2, 3, 4, 5, 6)
\end{aligned}
$$

The negation of S is (perfect, high, ..., none). The negation of sk is sk' = s(6-k), i.e., s5' = s1. The following rule is used to break ties: If max {di} = dh = dk, compare the next-min[max(wj',cjh)] with the next-min[max(wj',cjk)]. If these also tie, continue to third minimums, and so forth.

Additional discussion of these decision methods and the results of experiments applying them to Navy decision problems can be found in Larsen (1989). Their application to the network selection problem is discussed in the next section.

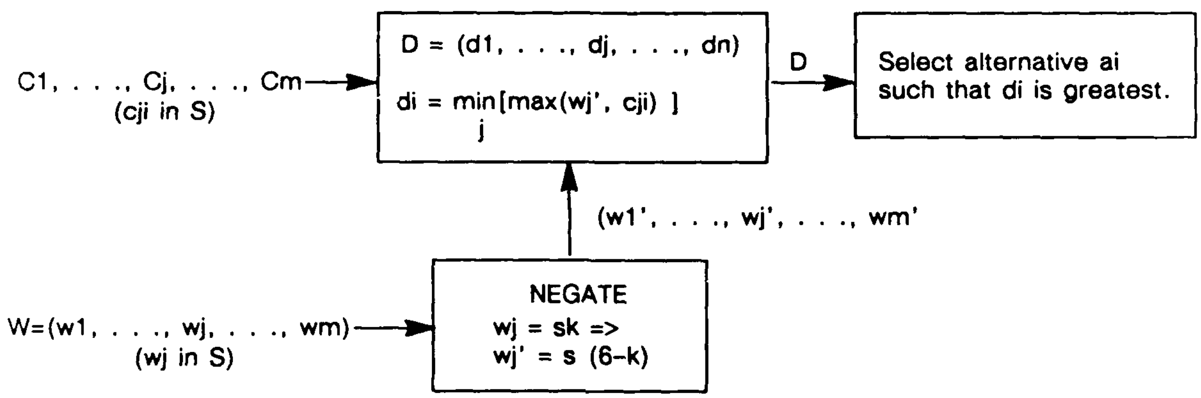

Figure 4. Yager (1981) version of the Max-Min method.

4.2. APPLICATION TO THE SELECTION PROBLEM

Figure 5 illustrates how the fuzzy decision methods described in the previous section can be applied to the network selection problem. Recall that measure cij indicates how well the jth alternative satisfies the ith criterion. Criteria weights are used with the Saaty (1970)/Yager (1970) and Yager (1981) methods.

We envision that if we decide to use fuzzy decision methods, an expert system would be used to formulate the appropriate decision problem, assign values to the fuzzy subsets, and perform the calculations. Algorithmic procedures would first be applied to eliminate inappropriate subnets, and the fuzzy algorithm would probably replace the decision statistics (e.g., the Point-to-Point Value Function described in the next section) currently proposed for the various message situations.

The main difficulty with applying fuzzy decision methods is that of assigning values of the measures cij (how well alternative i satisfies criteria j). Figure 6 gives two simple examples of how one might define a cost function for timeliness. Another difficulty would be the assigning of weights, since a consensus of users is not likely.

Assigning cost functions and weights are somewhat equivalent problems: neither the user nor the research community has any basis for proposing them, pending system level simulation. These problems will be investigated in the FY89 IED effort.

ALTERNATIVES: A = (Network 1, Network 2, Network 3, Network 4)

CRITERIA:
1. probability of connection/delivery on first try
2. timeliness (time to connection if connect first try)
3. balanced network loading
4. voice quality / error rate
5. security (LPI, jam/spoof resistance, etc.)

FUZZY MEASURES:

	Network				
	1	2	3	4	
Prob. Connect	(c11	c12	c13	c14)	= C1
Timeliness	(c21	c22	c23	c24)	= C2
Balance	(c31	c32	c33	c34)	= C3
Quality	(c41	c42	c43	c44)	= C4
Security	(c51	c52	c53	c54)	= C5

Figure 5. An application of Max-Min decision methods to a network selection problem.

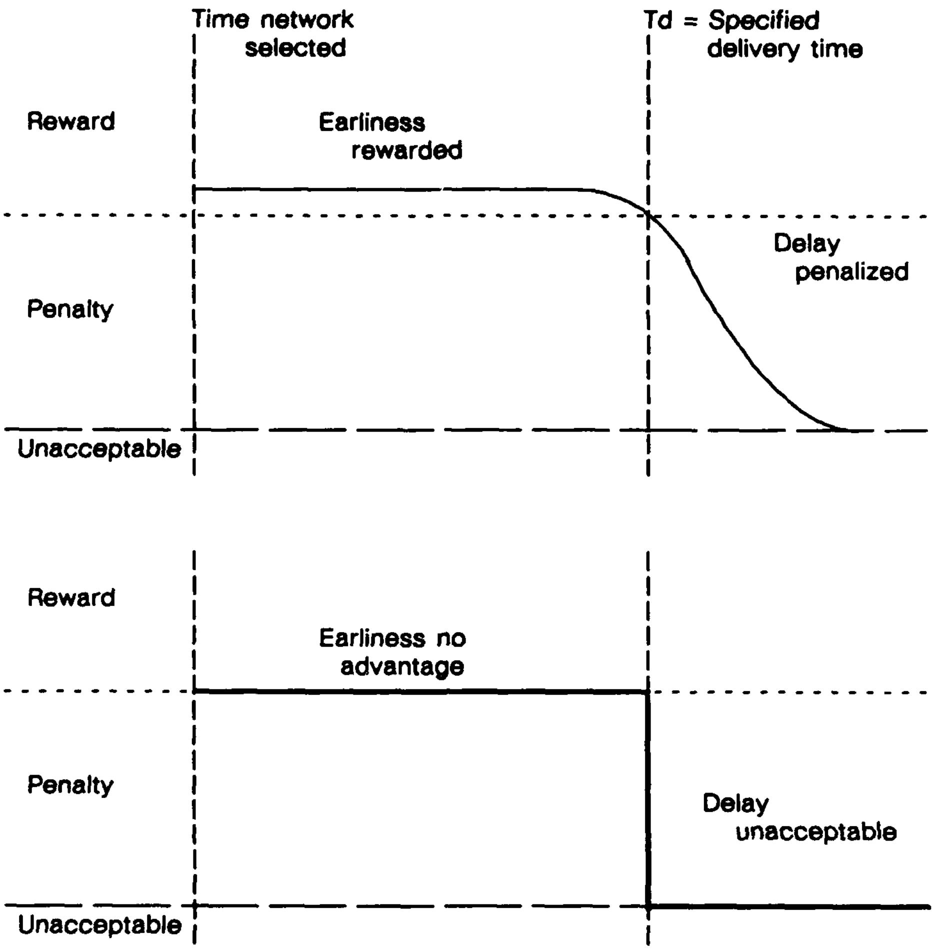

Figure 6. Example of timeliness cost functions.

5. APPLICABILITY OF COST/VALUE FUNCTIONS

5.1. BACKGROUND

The original focus of this IED as proposed by S. Norvell was to develop a cost function that could be used to solve the subnet selection problem. The scope of investigation has been enlarged, but the idea of using a cost or value function (cost functions assign relative penalties to performance parameters, value functions assign relative benefits) is still utilized as a subroutine in the selection process. Given that feasible, available subnets have been identified to service a particular Transmit Service Request (TSR), and more than one subnets can supply the minimum specified service parameters, a cost or value function is one way to make a selection between the subnets. There are other techniques such as linear or nonlinear optimization. Until the relative effects of various performance parameters are known, a cost function is a relatively simple final choice mechanism.

5.2. EXAMPLE VALUE FUNCTION

Presented here as an example is the value function developed for tie-breaking point-to-point datagram service requests. There are, at most, two subnets available; a precondition to the use of this value function is that both of these subnets pass all minimum performance threshold tests.

w_1P(Delivery) + w_2(Cp-Congestion)/Cp + w_3 (T-Message Delay)/T + w_4(Unt Priority-1)/Subnet Preference Class).

5.3. VALUE FUNCTION PARAMETERS

The most important attribute of the value function parameters is the range of values that can be attained *relative to the other parameters*. Each subnet is evaluated by the same function, so the contribution of a single parameter is negligible unless the parameter value has a relatively large range. Take Congestion for example: assume UHF LOS receives a 0.71 and HF ITF a 0.56 (before weighting). The difference is 0.15; after weighting, the difference becomes 0.30. Other things being equal, this will only make a difference in the final selection if the high weight Probability of Delivery parameters for the two networks are very close: differing by 0.03 or less.

The datagram value function is designed to emphasize the Probability of Delivery parameter for high priority messages, and Preference Class for low priority messages. The function is modified slightly by Congestion and Delay, allowing these factors to determine the outcome when the primary parameters are fairly even.

The point-to-point voice value function is designed to emphasize the probability of completing a voice circuit (Probability of Delivery) while allowing Congestion and Preference Class to play a minor part.

5.3.1. Probability of Delivery

Parameter: P(Delivery)
Suggested Weight: $w_1 = 10$
Attainable Raw Values: [0..1]
Attainable Weighted Values: [0..10].

Probability of Delivery is dynamically determined from current subnet performance. It is the single most important parameter in the value function, and is assigned a high weight.

5.3.2. Congestion

Parameter: (Cp-Congestion)/Cp
Suggested Weight:
Datagram Requests: $w_2 = 2$
Voice Requests: $w_2 = 3$
Attainable Raw Values: [0..1]
Attainable Weighted Values: [0..2].

The Congestion parameter is determined from current subnet conditions. Its primary use is a threshold limit that must be met. However, there is value to using the least congested subnet. The congestion parameter is normalized by taking the difference between the *allowed* congestion on a given subnet for a given priority and the *current* congestion on the subnet, then dividing by the allowed congestion. The resulting value is dependent both on the subnet's current state and the amount of data it can reasonably handle at the current node; normalizing this value allows comparison between subnets. A higher value reflects lower current congestion. This parameter is weighted higher than the delay parameter, since it is based on observed conditions. The parameter is weighted higher for voice requests (than datagram requests) since all buffered messages sit idle until termination of a voice call, creating added value to using a lightly congested subnet for voice services.

5.3.3. Delay

Parameter: (T-Delay)/T
Suggested Weight: $w_3 = 1$
Attainable Raw Values: [0..1]
Attainable Weighted Values: [0..1].

The delay parameter is the propagation delay while the message travels from source to destination. This is the minimum amount of time required to deliver the message. Queuing delays and delay while waiting for channel access is not included. (Research has determined that there is currently insufficient information to make any reasonable estimate of end-to-end delay until simulation and/or observed data is available.) This parameter is normalized in the same manner as Congestion; it is given a minimum weight, as it is not indicative of real delay over the subnet. A lower minimum delay results in a higher delay parameter value.

5.3.4. Preference Class

Parameter: (UNT Priority-1)/(Preference Class)
Suggested Weight: $w_4 = 1$
Attainable Raw Values
Numerator: {1, 2, 3, 4, 5, 6}
Denominator: {1, 2}
Attainable Weighted Values: {1/2, 1, 3/2, 2,..., 5, 11/2, 6}.

When selecting a subnet for a high priority message, performance is more important than Preference Class; for low priority messages, Preference Class is more important than subnet performance. This is reflected in the value function by dividing Unified Networking Technology (UNT) priority by Preference Class. The resulting contribution by the Preference Class parameter is large for low priority messages, and small for high priority messages.

6. APPLICABILITY OF EXPERT SYSTEMS

The need to investigate artificial intelligence solutions became evident several months into this project. In particular, the best option appeared to be to use an expert system shell. One reason the expert system approach to this problem is

appropriate is that the factual knowledge (also known as descriptive or declarative knowledge) can be easily and efficiently expressed in terms of frames compatible with the data structures of essentially all expert system shells. We will first consider the kinds of frames involved in this problem.

6.1. FACTUAL DOMAIN KNOWLEDGE

The simplest kind of frame consists of an object with attribute-value pairs, where the value of an attribute can sometimes be a list or other construct. The list below gives examples of frames at the higher level of domain factual knowledge. These frames were based on an earlier definition of Build 2, but are illustrative of the way frames would be constructed. The representation of data (e.g., under COMMPLAN, NTT, etc.) would vary with the type of shell used. Several options generally would be available, and the programmer would select the most efficient for the application.

Examples of Frames

```
UNT_internet
   build    build_2
   parts     (network user node)

network
   part_of          UNT_internet
   instances        (UHF_LOS HF_ITF)
   nodes            (adam ... ownnode)**
   capabilities     *
   status           *

   ** - value inherited by instances
   * - instances have different values for this property

UHF_LOS
   instance_of      network
   nodes            **
   capabilities     capab_UHF
   status           status_UHF

HF_ITF
   instance_of      network
   full_name        "High Frequency IntraTask Force"
   nodes            **
   capabilities     capab_HF
   status           status_HF

user
   part_of      UNT_Internet
   alias        subscriber
   instances    (NTDS NAVMACS voice)
   data_type    *

NTDS
   instance_of    user
   data_type      tactical_data
   parameter      <priority>

NAVMACS
   instance_of    user
   data_type    (record_messages file)

voice
   instance_of    user
   data_type    voice

node
   part_of      UNT_Internet
   alias        site
   instances    (adam  ...  ownnode)
   IP_address   *
   networks     (UHF_LOS HF_ITF)**
   users        (NTDS NAVMACS voice)**
```

```
adam
   instance_of   node
   site          <location or platform name>
   IP_address    192.9.200.XXX
   networks      **
   users         **

own_node
   instance_of   node
   site          <location or platform name>
   IP_address    193.9.600.XXX
   networks      **
   users         **
   parts         (MC UHF_LNC HF_LNC NTDS_SIU
                  NAVMACS_SIU voice_SIU)

MC
   part_of    own_node
   full_name  "Multi-Network Controller"
   site       <own_node's site>
   parts      (MC_database [data_manager] [network_selector])
```

[Bracketed items can be (i) objects having procedures, (ii) procedures, or (iii) rulesets. Likely to call outside processes or query outside databases, in embedded version.]

```
MC_database
   part_of  MC
   alias    "Network Administrator's database"
   parts     (TSR NW_capab_table NW_status_table COMMPLAN
             phonebook NTT NCRT packet_delay_stats best_path
             hops_table DCRT)

TSR
   part_of       MC_database
   full_name     "transmit service request"
   instances     (TSR_0001 TSR_0002 ... <current TSR>)
   source        *
   destination   *
   trans_mode    *
   security      *
   AJ            *
   bandwidth     *
   length        *
   data_type     *
   priority      *
   reliability   *
   timeliness    *
   options       *
```

```
TSR_0001
   instance_of   TSR
   source        NTDS_SIU
   destination   NTDS
   trans_node    broadcast
   security      GENSER/TS
   AJ            0
   bandwidth     0
   length        100
   data_type     NTDS
   priority      3

   :

NW_capab_table
   part_of       MC_database
   full_name     "Network Capabilities Table"
   variability   static
   parts         (capabilities_UHF capabilities_HF)

capabilities_UHF
   part_of          NW_capab_table
   network          UHF_LOS
   security         GENSER/TS
   AJ               modest
   bandwidth        4800
   length           600
   pt-pt_voice       1
   pt-pt_data        1
   .
   .
   brdcast_record   1

capabilities_HF
   part_of          NW_capab_table
   network          HF_ITF
   security         GENSER/TS
   AJ               none
   bandwidth        2400
   length           <TBD>
   pt-pt_voice       1
   pt-pt_data        1
   .
   .
   brdcast_record   1

NW_status_table
   part_of       MC_database
   variability   dynamic
   parts         (status_UHF status_HF)
```

```
status_UHF
   part_of              NW_status_table
   network              UHF_LOS
   equipment_status     1
   EMCON_status         1
   current_voice_use    0
   current_data_use     1
   current_file_use     0

status_HF
   part_of              NW_status_table
   network              HF_LOS
   equipment_status     1
   EMCON_status         1
   current_voice_use    0
   current_data_use     1
   current_file_use     0

COMPLAN
   part_of       MC_database
   variability   static
   used_by       pref_class_grouper
   ... <data or sub-objects with data > ...

phonebook
   part_of       MC_database
   variability   static
   used_by       address_translator
   ... <IP address data or sub-objects with data > ...

NTT
   part_of       MC_database
   full_name     "Network Topology Table"
   variability   dynamic
   built_by      updater/historian
   ... <data or sub-objects with data > ...

NCRT
   part_of       MC_database
   full_name     "Network Connectivity/Reliability Table"
   variability   dynamic
   built_by      performance predictor
   built_from    NTT
   ... <data or sub-objects with data > ...

packet_delay_stats
   part_of       MC_database
   variability   dynamic
   built_by      performance_predictor
   ... <data or sub-objects with data > ...
```

```
best_path
   part_of       MC_database
   variability   dynamic
   built_by      pathfinder
   built_from
   ... <data or sub-objects with data > ...

hops_table
   part_of       MC_database
   variability   dynamic
   built_by      pathfinder
   built_from
   ... <data or sub-objects with data > ...

DCRT
   part_of       MC_database
   full_name     "Destination Connectivity/Reliability Table"
   variability   dynamic
   built_by      performance_predictor
   built_from    (NCRT packet_delay_stats)
   ... <data or sub-objects with data > ...
```

6.2. PROCEDURAL DOMAIN KNOWLEDGE

In addition to factual domain knowledge, an expert system needs procedural knowledge. Some of the frames in the examples represent functions (e.g., the MC frame) and have procedures associated with them. Procedural knowledge includes knowledge about relationships among events and situations or states. In this application, the procedural knowledge is very much algorithmic. The two main ways of implementing procedural knowledge in an expert system are rule-oriented programming and object-oriented programming. In the latter case, the object (in an expanded sense) is a package of information, with individual procedures or behaviors attached to the object. The action in pure object-oriented programming results from message passing among the objects. The object-oriented capability is highly desirable if simulation is involved.

Figure 7 shows some of the objects (from the object-attribute-value frame examples) as a hierarchy. The rectangular objects represent those tied to procedural knowledge. The higher levels of frames would be implemented if steps 1 to 4 were integrated into a single expert system. Even if only step 4 is implemented in an expert system shell, some of these frames could be coded to help explain to a new user of the system how it works.

6.3. SELECTION OF AN EXPERT SYSTEM SHELL

Expert system shells tend to be rule based (i.e., rule oriented), object oriented (in the sense discussed earlier), or a hybrid combination. After considering the shells available, we selected the rule-based system 'C' Language Production System (CLIPS) developed at the NASA/JOHNSON Space Center. At the time we made the selection, no specific rules for step 4 had been formulated, but it was clear that many could be. The lack of simulation capability was not an important factor, since alternatives are available for providing representative input messages and the data derived from them. Our main reasons for selecting CLIPS were that it is (a) free to U.S. Government agencies, (b) highly portable, and (c) easily integrated with external systems. CLIPS also has a good set of debugging tools. It does not have inheritance mechanisms (which would allow offspring objects to inherit attributes and values of attributes from parent objects), but we did not consider inheritance important in this application. In the next section, we discuss CLIPS and its application to this problem in detail.

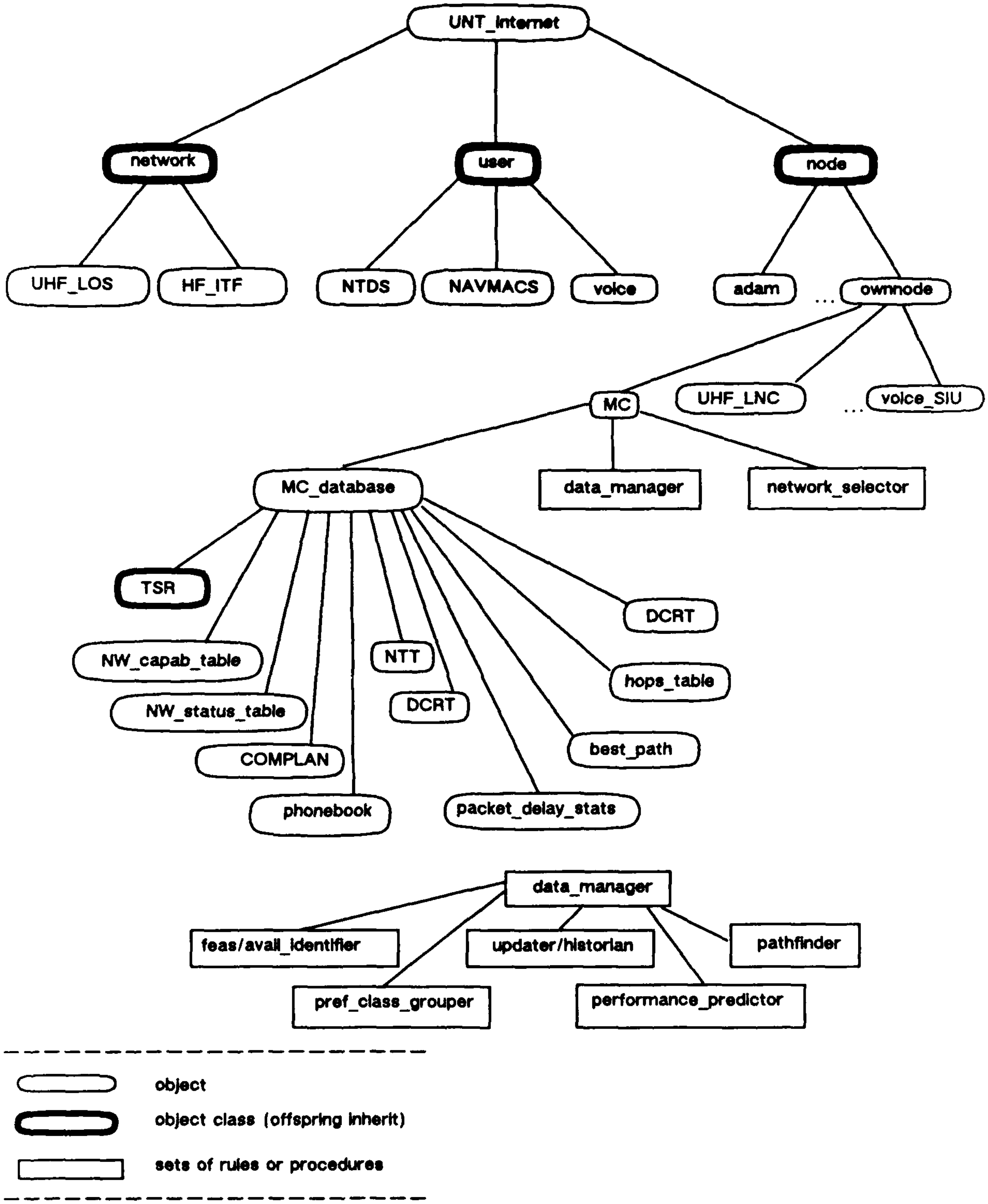

Figure 7. Hierarchical representation of objects.

6.3.1. Experiments in CLIPS

We have begun the coding of final selection rules in version 4.2 of the C Language Integrated Production System (CLIPS). CLIPS was developed by the Artificial Intelligence Section of the Mission Planning and Analysis Division at NASA/Johnson Space Center. Development of versions 4.1, 4.11 and 4.2 was jointly funded by NASA and the USAF. CLIPS is fully documented in references (Giarratano, April 1988 and Giarratano, June 1988).

6.3.2. Overview of CLIPS

CLIPS is written in C for portability and speed, and has been installed on many computers from different vendors with no code changes. It can be embedded within procedural code, called as a subroutine, and integrated with languages other than C, such as FORTRAN and ADA.

The basic elements of CLIPS are:

1. Fact-list: global memory for data. The function "deffacts" is used to create a list of facts.

2. Knowledge-base: contains all the rules. The function "deffrule" is used to create a rule.

3. Inference engine: decides which rules should be executed and controls overall execution.

CLIPS provides an interactive development environment, including debugging aids, on-line help, and an integrated editor.

6.3.3. Initial Implementation

The coding in CLIPS of subnet selection rules began with the point-to-point datagram case. The Appendix gives a typescript of the initial experiment, and the CLIPS code of the rules and data. The rules are for testing subnets for timeliness, congestion, probability of delivery, and joint probability of delivery. The rules are written in a much more general form than is currently needed for selecting between only two networks (UHF-LOS and HF- ITF), because we plan to extend the experiments to as many as 12 networks. The set of rules listed in the Appendix stop just short of computing the Point-to-Point Value Function for multicandidate situations. While this calculation will be performed, an alternative fuzzy decision algorithm will also be formulated and experimentally compared.

The early experiments will substitute canned data for simulation. The first rules listed in the Appendix control the consecutive reading in of files, one file per Transmit Service Request (TSR). The file includes the data given in the TSR and also the data that would be derived in steps 1 to 3 of the processing. The Appendix gives the file for the first TSR as an example. Other files will vary in only a few values of the derived data. The values used in the first seven experimental cases are given in Table 1. The values chosen are not necessarily realistic, but they have the desirable relationships between measurements and thresholds.

Most of the rules are applicable to more than point-to-point datagram requests. As rules for other situations are added, the more general rules will be organized into separate modules.

Table 1. Experimental point-to-point datagram data

	TSR = Timeliness = 90		congestion thr. = 7		TSR: reliability = 0.5	
	message_delay		congestion		prob_delivery	
TSR_i	UHF	HF	UHF	HF	UHF	HF
1	100	100	5	8	0.5	0.4
2	80	80	5	8	0.5	0.4
3	80	100	5	5	0.5	0.4
4	80	80	5	5	0.2	0.2
5	80	100	5	5	0.4	0.4
6	80	80	5	5	0.6	0.4
7	80	80	5	5	0.6	0.6

7. FISCAL YEAR 1989 PLANS

There are two projects funded for FY89 that pertain to the research begun in this IED project.

7.1 COST METRIC ALGORITHMS FOR INTERNETWORK APPLICATIONS ZE68 FOLLOW-ON

Principal Investigator: Robin Dillard, Code 444

Coding of selection rules will continue. The CLIPS program will first be tested using a simple simulation of the communication environment. After initial development, we plan to move the system to Code 854's NASTEE (Network Architecture Simulated Test and Evaluation Environment) testbed for further refinement, testing, and evaluation in a comprehensive simulation environment. The resulting system will

be a network control program that operates in a scaled-down simulated communication environment, demonstrating the potential of the decision algorithms or strategies in the real environment. Determination of the tradeoffs among efficiency, complexity, stability, etc., will enable expansion into a more complete system.

The plans for FY89 can be summarized as follows.

Phase 1. Development and Proof of Concept:

- Build rulesets for several networking models.
- Code rules in CLIPS.
- Test, select, and refine with simulated data.

Phase 2 Options:

- Embed CLIPS system in NASTEE/A for further testing and refining.
- Code final Phase-1 rules for integration with UNT ATD software.

The networking models referred to in phase 1 plans are summarized in Table 2.

7.2 INTERNETWORK ROUTING FOR MOBILE PACKET RADIO NETWORKS IR PROJECT 854-ZW11

Principal Investigator: David Olsen, Code 854

The FY88 IED reported in this technical note concentrated on methods of solving the subnet selection problem. This FY89 IR project tackles the larger problem of routing messages through the future CSS architecture (G. Brown, 1988). The current problem can be stated as: which subnet(s) will do the best job of delivering the message to the destination(s)? This assumes the destination(s) are known, and are reachable on one of the available subnets! The future problem can be stated as: Which subnet(s) will do the best job of delivering the message to the destination(s), or which subnet(s) can provide the best service to either (1) a gateway which can reach the destination(s), or (2) a node which can resolve an address conflict? This is a much larger problem! Measures of performance, and methods of estimating and disseminating those measures that are consistent with the CSS architecture (G. Brown, 1988), must be devised. An addressing scheme extendable to a global domain must be determined. Techniques of address resolution are involved: what does the internetwork routing algorithm do when one or more of the destinations are unknown? Once these issues are resolved or postulated, the techniques developed in this IED become applicable to the final subnet selection problem.

Table 2. Networking models

MODEL 1:			
UNT Internet		2 Networks	(UHF LOS, HF ITF)
	Build 2	3 Users	(Voice, NAVMACS, NTDS)
		7 Nodes	(5 shore sites, 1 ship, 1 aircraft)
		No internetwork gatewaying	
MODEL 2:			
UNT Internet		2 Networks	(UHF LOS, HF ITF)
	Build 3	3 Users	(Voice, NAVMACS, NTDS)
		7 Nodes	(5 shore sites, 1 ship, 1 aircraft)
		Internetwork gatewaying implemented	
MODEL 3:			
Small scale		3 Networks	
future UNT		4 Users	
environment		11 Nodes	
		Internetwork gatewaying implemented	
MODEL 4:			
Large scale		12 Networks	
future UNT		15 Users	
environment		40 Nodes	
		Internetwork gatewaying implemented	

REFERENCES

Bellman, R. E. and Zadeh, L. A., "Decision Making in a Fuzzy Environment," *Management Science, vol. 17,* pp. 141-164, 1970.

Brown, G., "Communication Support System (CSS) Integrated Satcom," Code 854 Naval Ocean Systems Center Internal Report, January 1988.

Code 824, "Advanced Technology Demonstration Unified Networking Technology Phase 1 Program Plan," Naval Ocean Systems Center, 30 January 1987.

Genest, C. and Zidek, J. V., "Combining Probability Distributions: A Critique and Annotated Bibliograph," *Statistical Science, vol. 1, no. 1,* pp. 114-134, 1986.

Giarratano, J. C., "CLIPS Reference Manual, Version 4.2 of CLIPS," Artificial Intelligence Section, Lyndon B. Johnson Space Center, April 1988.

Giarratano, J. C., "CLIPS User's Guide, Version 4.2 of CLIPS," Artificial Intelligence Section, Lyndon B. Johnson Space Center, June 1988.

Larsen, R. W., "A Fuzzy Logic Decision Tool," Naval Ocean Systems Center Technical Report (in preparation), 1989.

Olsen, D., "Multinetwork Controller Architecture Specification," Code 854 Naval Ocean Systems Center, 2 August 1988.

Saaty, T. L.,"A Scaling Method for Priorities in Hierarchical Structures," *Jour. of Mathematical Psychology, vol. 15,* pp. 234-281, 1977.

Yager, R. R., "Multiple Objective Decision Making Using Fuzzy Sets," *International Journal of Man-Machine Studies, vol. 9,* pp. 375-382, 1977.

Yager, R. R., "A New Methodology for Ordinal Multiobjective Decisions Based on Fuzzy Sets," *Decision Sciences, vol. 12,* no. 4, pp. 589-600, October 1981.

APPENDIX A

INITIAL EXPERIMENTS IN CLIPS

INITIAL EXPERIMENTS IN CLIPS

TYPESCRIPT:

```
CLIPS> (batch "start")
CLIPS> ;File "start" loads other files into CLIPS and asserts initial facts:
(open "robind/zrules1" zrules1)
1
CLIPS> (open "robind/zrulesa" zrulesa)
1
CLIPS> (open "robind/zrulesb" zrulesb)
1
CLIPS> (load "robind/zrules1")
*****
CLIPS> (load "robind/zrulesa")
***************
CLIPS> (load "robind/zrulesb")
*************************
CLIPS> (assert (initial-fact)
         (subnet UHF_LOS)
         (subnet HF_ITF))
CLIPS> (run)
Enter (integer) of next message:
1
How many messages do you want to process?
7
2 rules fired
CLIPS> ;First Transmit Service Request

 (assert
    ("TSR_1" message_ID 1)
    ("TSR_1" source x)
    ("TSR_1" destination y)
    ("TSR_1" trans_mode pt-to-pt)
    ("TSR_1" security GENSER/TS)
    ("TSR_1" AJ 0)
    ("TSR_1" bandwidth 0)
    ("TSR_1" length 20)
    ("TSR_1" data_type datagram)
    ("TSR_1" priority 2)
    ("TSR_1" reliability 0.5)
    ("TSR_1" timeliness 90))
 CLIPS>
 ;Derived data from steps 1-3 for first TSR

 (assert
    ("TSRdd_1" UNT_priority 2)
    ("TSRdd_1" pref_class_1 UHF_LOS)
    ("TSRdd_1" pref_class_2 HF_ITF)
    ("TSRdd_1" prob_deliv UHF_LOS 0.5)
    ("TSRdd_1" prob_deliv HF_ITF 0.4)
    ("TSRdd_1" message_delay UHF_LOS 100)
    ("TSRdd_1" message_delay HF_ITF 100)
    ("TSRdd_1" congestion UHF_LOS  5)
    ("TSRdd_1" congestion HF_ITF 8)
    ("TSRdd_1" max_congestion UHF_LOS 7)
    ("TSRdd_1" max_congestion HF_ITF 7))
 CLIPS> (run)
 Service on "TSR_1" is refused:
    UHF_LOS is untimely although not congested.
    HF_ITF is congested and untimely.
```

```
25 rules fired
CLIPS>

(assert
   ("TSR_2" message_ID 2)
   ("TSR_2" source x)
   ("TSR_2" destination y)
   ("TSR_2" trans_mode pt-to-pt)
   ("TSR_2" security GENSER/TS)
   ("TSR_2" AJ 0)
   ("TSR_2" bandwidth 0)
   ("TSR_2" length 20)
   ("TSR_2" data_type datagram)
   ("TSR_2" priority 2)
   ("TSR_2" reliability 0.5)
   ("TSR_2" timeliness 90))
CLIPS>
(assert
   ("TSRdd_2" UNT_priority 2)
   ("TSRdd_2" pref_class_1 UHF_LOS)
   ("TSRdd_2" pref_class_2 HF_ITF)
   ("TSRdd_2" prob_deliv UHF_LOS 0.5)
   ("TSRdd_2" prob_deliv HF_ITF 0.4)
   ("TSRdd_2" message_delay UHF_LOS 80)
   ("TSRdd_2" message_delay HF_ITF 80)
   ("TSRdd_2" congestion UHF_LOS  5)
   ("TSRdd_2" congestion HF_ITF 8)
   ("TSRdd_2" max_congestion UHF_LOS 7)
   ("TSRdd_2" max_congestion HF_ITF 7))
CLIPS> (run)
Dropped as candidate subnet(s):
   HF_ITF is congested although timely.
Subnet UHF_LOS is the only timely and uncongested subnet.
Next checking its reliability.  ...

Assign subnet UHF_LOS to "TSR_2".
   It is the only subnet to pass the timeliness, congestion, and
   probability of delivery tests.
30 rules fired
CLIPS>

(assert
   ("TSR_3" message_ID 3)
   ("TSR_3" source x)
   ("TSR_3" destination y)
   ("TSR_3" trans_mode pt-to-pt)
   ("TSR_3" security GENSER/TS)
   ("TSR_3" AJ 0)
   ("TSR_3" bandwidth 0)
   ("TSR_3" length 20)
   ("TSR_3" data_type datagram)
   ("TSR_3" priority 2)
   ("TSR_3" reliability 0.5)
   ("TSR_3" timeliness 90))
CLIPS>
(assert
   ("TSRdd_3" UNT_priority 2)
   ("TSRdd_3" pref_class_1 UHF_LOS)
   ("TSRdd_3" pref_class_2 HF_ITF)
   ("TSRdd_3" prob_deliv UHF_LOS 0.5)
   ("TSRdd_3" prob_deliv HF_ITF 0.4)
   ("TSRdd_3" message_delay UHF_LOS 80)
   ("TSRdd_3" message_delay HF_ITF 100)
   ("TSRdd_3" congestion UHF_LOS  5)
   ("TSRdd_3" congestion HF_ITF 5)
   ("TSRdd_3" max_congestion UHF_LOS 7)
```

```
   ("TSRdd_3" max_congestion HF_ITF 7))
CLIPS> (run)
Dropped as candidate subnet(s):
   HF_ITF is untimely although not congested.
Subnet UHF_LOS is the only timely and uncongested subnet.
Next checking its reliability.  ...

Assign subnet UHF_LOS to "TSR_3".
   It is the only subnet to pass the timeliness, congestion, and
   probability of delivery tests.
30 rules fired
CLIPS>

(assert
   ("TSR_4" message_ID 4)
   ("TSR_4" source x)
   ("TSR_4" destination y)
   ("TSR_4" trans_mode pt-to-pt)
   ("TSR_4" security GENSER/TS)
   ("TSR_4" AJ 0)
   ("TSR_4" bandwidth 0)
   ("TSR_4" length 20)
   ("TSR_4" data_type datagram)
   ("TSR_4" priority 2)
   ("TSR_4" reliability 0.5)
   ("TSR_4" timeliness 90))
CLIPS>
(assert
   ("TSRdd_4" UNT_priority 2)
   ("TSRdd_4" pref_class_1 UHF_LOS)
   ("TSRdd_4" pref_class_2 HF_ITF)
   ("TSRdd_4" prob_deliv UHF_LOS 0.2)
   ("TSRdd_4" prob_deliv HF_ITF 0.2)
   ("TSRdd_4" message_delay UHF_LOS 80)
   ("TSRdd_4" message_delay HF_ITF 80)
   ("TSRdd_4" congestion UHF_LOS  5)
   ("TSRdd_4" congestion HF_ITF 5)
   ("TSRdd_4" max_congestion UHF_LOS 7)
   ("TSRdd_4" max_congestion HF_ITF 7))
CLIPS>
(run)
No candidate network is untimely or congested.
More than one candidate subnet.
Next checking their reliability.  ...

Joint probability of delivery for HF_ITF and UHF_LOS is 0.36000001.
Service on "TSR_4" is refused:
   No subnet or combination of subnets has a sufficiently high
   probability of successful delivery.
31 rules fired
CLIPS>


(assert
   ("TSR_5" message_ID 5)
   ("TSR_5" source x)
   ("TSR_5" destination y)
   ("TSR_5" trans_mode pt-to-pt)
   ("TSR_5" security GENSER/TS)
   ("TSR_5" AJ 0)
   ("TSR_5" bandwidth 0)
   ("TSR_5" length 20)
   ("TSR_5" data_type datagram)
   ("TSR_5" priority 2)
   ("TSR_5" reliability 0.5)
   ("TSR_5" timeliness 90))
```

```
CLIPS>
(assert
   ("TSRdd_5" UNT_priority 2)
   ("TSRdd_5" pref_class_1 UHF_LOS)
   ("TSRdd_5" pref_class_2 HF_ITF)
   ("TSRdd_5" prob_deliv UHF_LOS 0.4)
   ("TSRdd_5" prob_deliv HF_ITF 0.4)
   ("TSRdd_5" message_delay UHF_LOS 80)
   ("TSRdd_5" message_delay HF_ITF 100)
   ("TSRdd_5" congestion UHF_LOS  5)
   ("TSRdd_5" congestion HF_ITF 5)
   ("TSRdd_5" max_congestion UHF_LOS 7)
   ("TSRdd_5" max_congestion HF_ITF 7))
CLIPS> (run)
Dropped as candidate subnet(s):
   HF_ITF is untimely although not congested.
Subnet UHF_LOS is the only timely and uncongested subnet.
Next checking its reliability.  ...

Service on "TSR_5" is refused:
   Only one subnet passed the timeliness and congestion tests, and
   it failed the probability of delivery test.
30 rules fired
CLIPS>

(assert
   ("TSR_6" message_ID 6)
   ("TSR_6" source x)
   ("TSR_6" destination y)
   ("TSR_6" trans_mode pt-to-pt)
   ("TSR_6" security GENSER/TS)
   ("TSR_6" AJ 0)
   ("TSR_6" bandwidth 0)
   ("TSR_6" length 20)
   ("TSR_6" data_type datagram)
   ("TSR_6" priority 2)
   ("TSR_6" reliability 0.5)
   ("TSR_6" timeliness 90))
CLIPS>
(assert
   ("TSRdd_6" UNT_priority 2)
   ("TSRdd_6" pref_class_1 UHF_LOS)
   ("TSRdd_6" pref_class_2 HF_ITF)
   ("TSRdd_6" prob_deliv UHF_LOS 0.6)
   ("TSRdd_6" prob_deliv HF_ITF 0.4)
   ("TSRdd_6" message_delay UHF_LOS 80)
   ("TSRdd_6" message_delay HF_ITF 80)
   ("TSRdd_6" congestion UHF_LOS 5)
   ("TSRdd_6" congestion HF_ITF 5)
   ("TSRdd_6" max_congestion UHF_LOS 7)
   ("TSRdd_6" max_congestion HF_ITF 7))
CLIPS>
(run)
No candidate network is untimely or congested.
More than one candidate subnet.
Next checking their reliability.  ...

Assign subnet UHF_LOS to "TSR_6".
   It is the only subnet to pass the timeliness, congestion, and
   probability of delivery tests.
29 rules fired
CLIPS>

(assert
   ("TSR_7" message_ID 7)
   ("TSR_7" source x)
```

```
   ("TSR_7" destination y)
   ("TSR_7" trans_mode pt-to-pt)
   ("TSR_7" security GENSER/TS)
   ("TSR_7" AJ 0)
   ("TSR_7" bandwidth 0)
   ("TSR_7" length 20)
   ("TSR_7" data_type datagram)
   ("TSR_7" priority 2)
   ("TSR_7" reliability 0.5)
   ("TSR_7" timeliness 90))
CLIPS>
(assert
   ("TSRdd_7" UNT_priority 2)
   ("TSRdd_7" pref_class_1 UHF_LOS)
   ("TSRdd_7" pref_class_2 HF_ITF)
   ("TSRdd_7" prob_deliv UHF_LOS 0.6)
   ("TSRdd_7" prob_deliv HF_ITF 0.6)
   ("TSRdd_7" message_delay UHF_LOS 80)
   ("TSRdd_7" message_delay HF_ITF 80)
   ("TSRdd_7" congestion UHF_LOS  5)
   ("TSRdd_7" congestion HF_ITF 5)
   ("TSRdd_7" max_congestion UHF_LOS 7)
   ("TSRdd_7" max_congestion HF_ITF 7))
CLIPS>
(run)
No candidate network is untimely or congested.
More than one candidate subnet.
Next checking their reliability.  ...

At this point, the value of the point-to-point value function
 is computed for the two candidate nets.  This part hasn't been
 coded yet.
24 rules fired
CLIPS> (dribble-off)
```

FILES:

```
;File "start" loads other files into CLIPS and asserts initial facts:
(open "robind/zrules1" zrules1)
(open "robind/zrulesa" zrulesa)
(open "robind/zrulesb" zrulesb)
(load "robind/zrules1")
(load "robind/zrulesa")
(load "robind/zrulesb")
(assert (initial-fact)
        (subnet UHF_LOS)
        (subnet HF_ITF))
;First Transmit Service Request
------------

(assert
   ("TSR_1" message_ID 1)
   ("TSR_1" source x)
   ("TSR_1" destination y)
   ("TSR_1" trans_mode pt-to-pt)
   ("TSR_1" security GENSER/TS)
   ("TSR_1" AJ 0)
   ("TSR_1" bandwidth 0)
   ("TSR_1" length 20)
   ("TSR_1" data_type datagram)
   ("TSR_1" priority 2)
   ("TSR_1" reliability 0.5)
   ("TSR_1" timeliness 90))
```

```
;Derived data from steps 1-3 for first TSR

(assert
   ("TSRdd_1" UNT_priority 2)
   ("TSRdd_1" pref_class_1 UHF_LOS)
   ("TSRdd_1" pref_class_2 HF_ITF)
   ("TSRdd_1" prob_deliv UHF_LOS 0.5)
   ("TSRdd_1" prob_deliv HF_ITF 0.4)
   ("TSRdd_1" message_delay UHF_LOS 100)
   ("TSRdd_1" message_delay HF_ITF 100)
   ("TSRdd_1" congestion UHF_LOS  5)
   ("TSRdd_1" congestion HF_ITF 8)
   ("TSRdd_1" max_congestion UHF_LOS 7)
   ("TSRdd_1" max_congestion HF_ITF 7))

-------


;DATA LOADING RULES

;       [For early testing of network-assignment rules.  Later versions
;        will read data and create the facts now in files.]


(defrule user_input
   ?a<-(initial-fact)
=>
   (fprintout t "Enter (integer) of next message:" crlf)
   (bind ?num (read))
   (assert (currentnum ?num))
   (bind ?name (str_cat robind/TSR_ ?num))
   (assert (currentfile ?name))     ; Has quotes around it
   (bind ?logicname (str_cat m ?num))
   (assert (currentlogic ?logicname))
   (fprintout t "How many messages do you want to process?" crlf)
   (bind ?total (read))
   (assert (lastmsgnum =(+ ?total ?num -1))
           (load message))
   (retract ?a))



(defrule readmessage
   ?a<-(load message)
   (currentnum ?num)
   (currentfile ?name)
   (currentlogic ?logicname)
=>
   (bind ?tsr (str_cat TSR_ ?num))
   (assert (current_TSR ?tsr))  ;Has quotes around it
   (bind ?tsrdd (str_cat TSRdd_ ?num))
   (assert (current_TSRdd ?tsrdd))  ;Has quotes around it
   (open ?name ?logicname)
   (batch ?name)
   (assert (process message))
   (retract ?a))

    ; MESSAGE PROCESSING RULES fire, asserting (cleanup) when done.

(defrule cleanup
   ?a<-(cleanup)
   ?cf<-(currentfile ?name)
   ?cl<-(currentlogic ?logicname)
   ?ct<-(current_TSR  ?TSR)
   ?cd<-(current_TSRdd ?TSRdd)
```

```
=>
   (close ?logicname)
   (assert (message done))
   (retract ?a ?cf ?cl ?ct ?cd))


(defrule no_more
   (message done)
   (currentnum ?num)
   (lastmsgnum ?lastnum)
   (test (= ?num ?lastnum))
=>
   (fprintout t "That was the last message." crlf)
   (halt))


(defrule next_message
   ?a<-(message done)
   ?b<-(currentnum ?num)
   (lastmsgnum ?lastnum)
   (test (< ?num ?lastnum))
=>
   (bind ?nextnum (+ 1 ?num))
   (assert (currentnum ?nextnum))
   (bind ?nextfile (str_cat robind/TSR_ ?nextnum))
   (bind ?logicname (str_cat m ?nextnum))
   (assert (currentfile ?nextfile)    ;  Has quotes around it
           (currentlogic ?logicname)
           (load message))
   (retract ?a ?b))
-------


;File: zrulesA (point-to-point datagrams - part A)


(defrule count_nets
   (process message)
   (current_TSRdd ?tsrdd)
   (?tsrdd pref_class_1 $?nets1)
   (?tsrdd pref_class_2 $?nets2)
=>
   (bind ?netcount1 (length $?nets1))
   (bind ?netcount2 (length $?nets2))
   (assert (netcount =(+ ?netcount1 ?netcount2))
           (pref1 $?nets1)
           (pref2 $?nets2)))

(defrule pt-pt_gram    "Preparation for point-to-point datagrams"
   ?a<-(process message)
   (netcount ?netcount)
   (current_TSR ?tsr)
   (?tsr trans_mode pt-to-pt)
   (?tsr data_type datagram)
=>
   (assert (current_categ pt-pt_gram)
           (initial_tests)
           (testcount_t ?netcount)
           (testcount_c ?netcount))
   (retract ?a))


(defrule test_t_c    "Test for timeliness and congestion."
   (initial_tests)
   (current_categ pt-pt_gram)
   (subnet ?net)
```

```
   (pref1 $?nets1)
   (pref2 $?nets2)
   (or (test (member ?net $?nets1))
       (test (member ?net $?nets2)))
=>
   (assert (test_timeliness ?net)
           (test_congestion ?net)))


;At least one preference group should have at least one subnet.


(defrule test_timeliness1
   ?x<-(test_timeliness ?net)
   (current_TSR ?tsr)
   (current_TSRdd ?tsrdd)
   (?tsr timeliness ?thr)
   (?tsrdd message_delay ?net ?delay)
=>
   (assert (compare_t ?net ?delay ?thr))
   (retract ?x))


(defrule test_timeliness2
   ?x<-(compare_t ?net ?delay ?thr)
   (test (<= ?delay ?thr))
   ?y<-(testcount_t ?count)
=>
   (assert (timely ?net)
           (testcount_t =(- ?count 1)))
   (retract ?x ?y))


(defrule test_timeliness3
   ?x<-(compare_t ?net ?delay ?thr)
   (test (> ?delay ?thr))
   ?y<-(testcount_t ?count)
=>
   (assert (untimely ?net)
           (testcount_t =(- ?count 1)))
   (retract ?x ?y))


(defrule test_congestion1
   ?x<-(test_congestion ?net)
   (current_TSRdd ?tsrdd)
   (?tsrdd congestion ?net ?congestion)
   (?tsrdd max_congestion ?net ?thr)
=>
   (assert (compare_c ?net ?congestion ?thr))
   (retract ?x))


(defrule test_congestion2
   ?x<-(compare_c ?net ?congestion ?thr)
   (test (<= ?congestion ?thr))
   ?y<-(testcount_c ?count)
=>
   (assert (uncongested ?net)
           (testcount_c = (- ?count 1)))
   (retract ?x ?y))


(defrule test_congested3
   ?x<-(compare_c ?net ?congestion ?thr)
   (test (> ?congestion ?thr))
```

```
   ?y<-(testcount_c ?count)
=>
   (assert (congested ?net)
           (testcount_c =(- ?count 1)))
   (retract ?x ?y))

(defrule stop_t_c_tests   "Stop testing for timeliness and congestion."
   ?a<-(initial_tests)
   ?b<-(testcount_t 0)
   ?c<-(testcount_c 0)
=>
   (assert (which passed)
           (passed_count 0)
           (failed_count 0))
   (retract ?a ?b ?c))

;Note that facts (pref1 ...) and (pref2 ...) remain.


(defrule passed_t_c_tests
   (which passed)
   ?x<-(timely ?net)
   ?y<-(uncongested ?net)
   ?z<-(passed_count ?count)
=>

   (assert (passed_count =(+ ?count 1))
           (passed_t_c_tests ?net))
   (retract ?x ?y ?z))

(defrule failed_c_test
   (which passed)
   ?x<-(timely ?net)
   ?y<-(congested ?net)
   ?z<-(failed_count ?count)
=>
   (bind ?newcount (+ ?count 1))
   (assert (failed_count ?newcount)
           (failed_c_test ?net))
   (retract ?x ?y ?z))

(defrule failed_t_test
   (which passed)
   ?x<-(untimely ?net)
   ?y<-(uncongested ?net)
   ?z<-(failed_count ?count)
=>
   (assert (failed_count =(+ ?count 1))
           (failed_t_test ?net))
   (retract ?x ?y ?z))

(defrule failed_t_c_tests
   (which passed)
   ?x<-(untimely ?net)
   ?y<-(congested ?net)
   ?z<-(failed_count ?count)
=>
   (assert (failed_count =(+ ?count 1))
           (failed_t_c_tests ?net))
   (retract ?x ?y ?z))
```

```
(defrule end_t_c_count
   ?a<-(which passed)
   (failed_count ?fc)
   (passed_count ?pc)
   (netcount ?netcount)
   (test (= ?netcount (+ ?fc ?pc)))
=>
   (assert (announce t_c_results))
   (retract ?a))
-------

;File zrulesB (point-to-point datagrams, part B)


(defrule all-failed_t_c
   ?a<-(announce t_c_results)
   (passed_count 0)
   (current_TSR ?tsr)
=>
   (fprintout t "Service on " ?tsr " is refused:" crlf)
   (assert (list failures))
   (retract ?a))


(defrule some_failed_t_c
   ?a<-(announce t_c_results)
   (not (passed_count 0))
   (not (failed_count 0))
=>
   (fprintout t "Dropped as candidate subnet(s):" crlf)
   (assert (list failures))
   (retract ?a))


(defrule none_failed_t_c    "Moves action to next rule set"
   ?a<-(announce t_c_results)
   (failed_count 0)
=>
   (fprintout t "No candidate network is untimely or congested." crlf)
   (assert (carryon))
   (retract ?a))


(defrule list_t_c_failures
   (list failures)
   (subnet ?net)
   ?f<-(failed_t_c_tests ?net)
=>
   (fprintout t "    " ?net " is congested and untimely." crlf)
   (retract ?f))


(defrule list_c_failures
   (list failures)
   (subnet ?net)
   ?f<-(failed_c_test ?net)
=>
   (fprintout t "    " ?net " is congested although timely." crlf)
   (retract ?f))


(defrule list_t_failures
   (list failures)
   (subnet ?net)
   ?f<-(failed_t_test ?net)
```

```
=>
   (fprintout t "    " ?net " is untimely although not congested." crlf)
   (retract ?f))


(defrule listed_t_c_failures
   ?a<-(list failures)
   (not (failed_t_test ?net1))
   (not (failed_c_test ?net2))
   (not (failed_t_c_tests ?net3))
=>
   (assert (listed failures))
   (retract ?a))


(defrule wind_down
   ?a<-(listed failures)
   (passed_count 0)
=>
   (assert (local_cleanup_t_c))
   (retract ?a))


(defrule local_cleanup_t_c
   ?a<-(local_cleanup_t_c)
   ?ty<-(current_categ pt-pt_gram)
   ?n1<-(pref1 $?nets1)    ;  Might move to "cleanup" if commmon
   ?n2<-(pref2 $?nets2)    ;    to all categories.
   ?p<-(passed_count ?pc)
   ?f<-(failed_count ?fc)
   ?n3<-(netcount ?nc)
=>
   (assert (cleanup))
   (retract ?a ?ty ?n1 ?n2 ?p ?f ?n3))


(defrule limp_on   "Rule after this fires if only 2 subnets."
   ?a<-(listed failures)
   (not (passed_count 0))
=>
   (assert (carryon))
   (retract ?a))


(defrule one_passed_t_c   "And-then-there-was-one"
   ?a<-(carryon)
   (passed_count 1)
   (passed_t_c_tests ?net)
   (current_TSR ?tsr)
=>
   (fprintout t  "Subnet " ?net " is the only timely and uncongested subnet."
        crlf "Next checking its reliability.  ..." crlf crlf)
   (assert (check probabilities)
           (passed_p_count 0)
           (check_count 0))
   (retract ?a))


(defrule some_passed_t_c   "Two (or more) passed time & congestion tests."
   ?a<-(carryon)
   (passed_count ?c)
   (test (> ?c 1))
=>
   (fprintout t "More than one candidate subnet." crlf
   "Next checking their reliability.  ..." crlf crlf)
   (assert (check probabilities)
```

```
          (passed_p_count 0)
          (check_count 0))
   (retract ?a))


(defrule prob_check1
   (check probabilities)
   ?x<-(passed_t_c_tests ?net)
   (current_TSR ?tsr)
   (current_TSRdd ?tsrdd)
   (?tsr reliability ?thr)
   (?tsrdd prob_deliv ?net ?prob)
=>
   (assert (compare_p ?net ?prob ?thr))
   (retract ?x))


(defrule prob_check2
   ?x<-(compare_p ?net ?prob ?thr)
   (test (>= ?prob ?thr))
   ?y<-(check_count ?cc)
   ?z<-(passed_p_count ?pc)
=>
   (assert (check_count =(+ ?cc 1))
          (passed_p_count =(+ ?pc 1))
          (passed_p_test ?net))
   (retract ?x ?y ?z))


(defrule prob_check3
   ?x<-(compare_p ?net ?prob ?thr)
   (test (< ?prob ?thr))
   ?y<-(check_count ?c)
=>
   (assert (check_count =(+ ?c 1))
          (failed_p_test ?net))
   (retract ?x ?y))


(defrule stop_prob_check
   ?a<-(check probabilities)
   (passed_count ?pc)   ;  Passed time and congestion tests
   (check_count ?cc)    ;  Were checked for reliability
   (test (= ?pc ?cc))
=>
   (assert (count successes))
   (retract ?a))


(defrule one_passed_p     "Another And-then-there-was-one rule"
   ?a<-(count successes)
   (passed_p_count 1)
   ?x<-(passed_p_test ?net)
   (current_TSR ?tsr)
=>
   (fprintout t "Assign subnet " ?net " to " ?tsr "." crlf
   "    It is the only subnet to pass the timeliness, congestion, and" crlf
   "    probability of delivery tests." crlf)
   (assert (local cleanup_p))
   (retract ?a ?x ))


(defrule no_hope
   ?a<-(count successes)    ; Just passed success probability test
   (passed_p_count 0)   ; Ealier passed time/congestion tests
   (passed_count 1)
```

```
   ?x<-(failed_p_test ?net)  ; part of local cleanup
   (current_TSR ?tsr)
=>
   (fprintout t "Service on " ?tsr " is refused:"  crlf
   "   Only one subnet passed the timeliness and congestion tests, and " crlf
   "   it failed the probability of delivery test."  crlf)
   (assert (local cleanup_p))
   (retract ?a ?x))


(defrule still_hope   "This rule assumes that only 2 subnets are available."
   ?a<-(count successes)
   (passed_p_count 0)
   ?x<-(failed_p_test ?net1)
   ?y<-(failed_p_test ?net2&:(neq ?net1 ?net2))
=>
   (assert (check_joint ?net1 ?net2))
   (retract ?a ?x ?y))


(defrule compute_joint_prob
   ?x<-(check_joint ?net1 ?net2)
   (current_TSRdd ?tsrdd)
   (?tsrdd prob_deliv ?net1 ?prob1)
   (?tsrdd prob_deliv ?net2 ?prob2)
=>
   (bind ?jp (- (+ ?prob1 ?prob2) (* ?prob1 ?prob2)))
   (fprintout t "Joint probability of delivery for " ?net1 " and " ?net2 " is "
    ?jp "." crlf)
   (assert (joint_prob ?net1 ?net2 ?jp))
   (retract ?x))


(defrule compare_joint_prob1
   ?x<-(joint_prob ?net1 ?net2 ?jp)
   (current_TSR ?tsr)
   (?tsr reliability ?thr)
   (test (< ?jp ?thr))
=>
   (fprintout t "Service on " ?tsr " is refused:"  crlf
   "   No subnet or combination of subnets has a sufficiently high" crlf
   "   probability of successful delivery." crlf)
   (assert (local cleanup_p))
   (retract ?x))


(defrule compare_joint_prob2
   ?x<-(joint_prob ?net1 ?net2 ?jp)
   (current_TSR ?tsr)
   (?tsr reliability ?thr)
   (test (>= ?jp ?thr))
=>
   (fprintout t "Assign both " ?net1 " and " ?net2 " to " ?tsr "." crlf)
   (assert (local cleanup_p))
   (retract ?x))


(defrule still_competing
   ?a<-(count successes)
   (passed_p_count 2)
   ?x<-(passed_p_test ?net1)
   ?y<-(passed_p_test ?net2&:(neq ?net1 ?net2))
=>
   (fprintout t "At this point, the value of the point-to-point value function"
    crlf " is computed for the two candidate nets.  This part hasn't been"
    crlf " coded yet." crlf)
```

```
   (retract ?a ?x ?y))


(defrule local_cleanup_p
   ?a<-(local_cleanup_p)
   ?x<-(passed_p_count ?pc)
   ?y<-(check_count ?cc)
=>
   (assert (local_cleanup_t_c))
   (retract ?a ?x ?y))
```